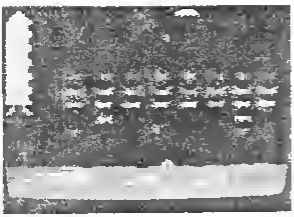
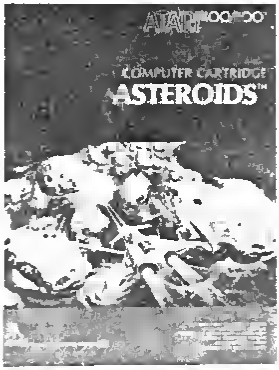
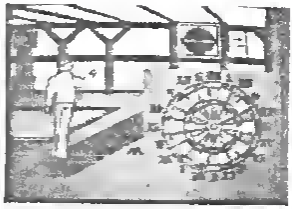
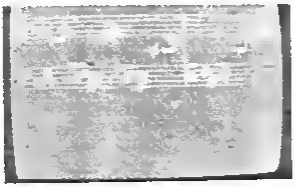
THE U.K. ATARI
COMPUTER OWNERS CLUB
(an independent users group)









Inverness
Aberdeen
Dundee
Perth
St Andrews
GLASGOW
EDINBURGH
Kilmarnock
NEWCASTLE
Carlisle
TEESSIDE
Darlington
Isle of Man
Douglas
Lancaster
Blackpool
BRADFORD
LEEDS
HULL
York
Grimsby
Bolton
MANCHESTER
LIVERPOOL
Anglesey
Chester
SHEFFIELD
Lincoln
STOKE on-TRENT
Derby
NOTTINGHAM
Shrewsbury
LEICESTER
Great Yarmouth
Lowestoft
Norwich
Wolverhampton
Peterborough
BIRMINGHAM
Northampton
Cambridge
Ipswich
Worcester
Hereford
Luton
Clacton on Sea
Gloucester
Oxford
Southend-on Sea
Swansea
CARDIFF
Swindon
Reading
GREATER LONDON
Margate
BRISTOL
Canterbury
Dover
Ashford
SOUTHAMPTON
PORTSMOUTH
Brighton
Hastings
Exeter
Bournemouth
Isle of Wight
PLYMOUTH

STAFF :-

Co-Editor---Graham Daubney

Co-Editor---Chris Earlow

Software Librarian---Ron levy

At last help for those of you who were about to scream and a chance for those of you who have discovered some secrets to share them with people who have a common interest,namely ATARI.

Firstly let me introduce the existing staff who will be working on this newsletter,look in the nearest mirror and there are the staff ! But someone had to get the ball rolling or should it be the keyboards clattering.The three of us decided that the time had come to get things started,after that its up to you to help by either contributions or by feedback to steer us in the direction you think we should be going.

The two editors of this quarterly newsletter will be myself and Chris,backed up by Ron who is managing the software side of things ,his introduction appears elsewhere in this issue(be sure to read it).

Both Chris and I ,although proffesionally involved are both enthusiasts and this entire publicaton has been produced in our own time.

I started getting involved in Micro's when the PET arrived and moved on through a bigger PET to an Apple ,via a SHARP MZ-80K and finally to ATARI.Chris has taken much the same route,but is perhaps more hardware orientated than myself(I once actually caught him holding a soldering iron next to an 8080 !).Ron is actively involved in the programming of our PDP-11 system at Maplin and his first small machine is the ATARI.

Enough about us,let me outline the ideas that we had in producing this first issue.

After much discussion it was decided that the ATARI machines had a lot of secrets locked up inside them that needed some explaination,also many of you would not necessarily have great experience in programming.The aim of this publication is to help,and it will contain articles on some of the more advanced techniques as well as explainations of some basic ideas.

I have been 'playing' with the ATARI computer systems for well over a year now and so have managed to pick up on many of its extended graphics capabilities,my aim will be to try and explain some of these facilities,firstly in a simple way and then as time goes by I will go back to a subject and cover it in greater depth.This issue starts with a basic explanation of how the character set is stored and how to write customised character sets.

In general this newsletter is designed to help,but for us to help you,some of you must help us and therefore we will gratefully recieve any articles or programs that you may care to contribute.Anything published in this newsletter will be public domain and neither us nor you will recieve any remuneration for our labours,however perhaps between us we can help each other to increase our knowledge and pleasure of computing.

Graham.J.Daubney

THIS ISSUE.

This issue has been produced by myself,Chris and Ron,with help from a friend of ours in Chichester,the first member and the first person on the software contributions map.

There will be no set features or hard and fast rules but a few things will make regular appearances,these are:-

THE SOFTWARE MAP

Ron is software librarian and although we will not publish all the programs that are sent in,we will publish a selection of what is in our opinion either,outstanding or likely to be of wide interest.Everyone who contibutes to the library will appear on our map.(unless of course he(or she ??) wishes to remain anonymous of course).

THE LIBRARY LISTING

This will hopefully grow and keep growing,it will appear each quarter until such time as it gets too large,it will then be issued as a listing in its own right and the quarterly newsletter will act as an update.

GRAPHICS

This will be written by Graham and will continue to appear until such time as all the various sections have been covered ie Scrolling,Player Missiles etc.It is hoped(mostly by Grahams wife) that guest writers will appear in this section from time to time.Many of the demonstration programs that will be used in this section are developed by Chris from Graham's text or thoughts,Chris being a batchelor type tends to have the most developement time.

PROGRAMS

We will endevour to print as many listings as we can,this will depend to a large extent on yourselves,so don't be shy ! It may not be the best thing you've ever written but someone somewhere will gain knowledge or pleasure from it,send it in and let us take a look at it.

PROBLEMS

You are invited to write in with your software problems or answers to the previous issue,again it will probably not be possible to publish all of them but we will do our best

CONTACT PAGE

This is very much your page where if you have a special interest you may look for fellow enthusiasts :-

Gentleman with player missile graphics problem seeks understanding young lady with player missile knowledge for mutual pleasure.....,you know the type of thing.

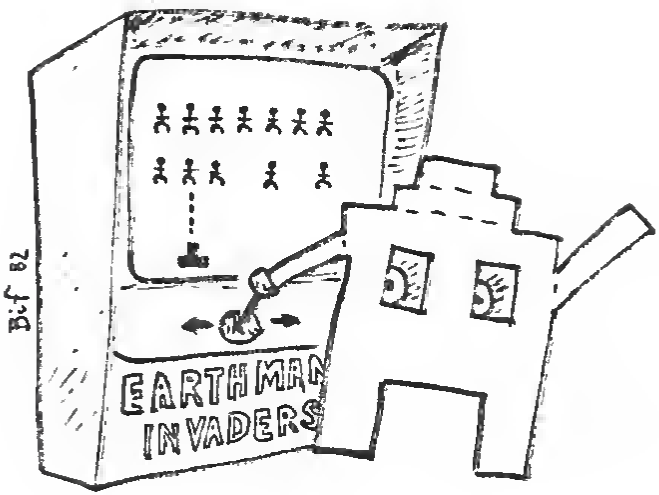
If anyone at this time feels like offering a regular feature or a 'series' then please contact us with the details and if possible the first instalment for our perusal.

CONTRIBUTIONS FORMAT

If any of put pen to paper or finger to keyboard,there are a few things that you can do to help us.

Firstly if possible 'type it',this will enable us to copy it directly and save greatly in our preperation time,it should be typed or printed on a standard A4 sheet of plain white paper.Each sheet of paper should have about an inch to an inch and a half of border at the top and should stop about an inch from the bottom,if you don't have a typwriter or printer then please write clearly and don't feel excluded as we will type up your work for you.

When sending in programs you don't have to worry too much about how you send it as we of course have access to all the necessary hardware to handle your media.The most important thing about sending in software is to make sure it works !! and to document it in some way,the best way to do this is with a short piece of text in REM statements at the beginning of your program or on an sheet of paper sent in with your program.We will accept programs in a listed format,however this does tend to wear Ron's fingers as he has to type them in to prove they work.The inner most workings of the Software exchange system are described by Ron elsewhere in this issue.



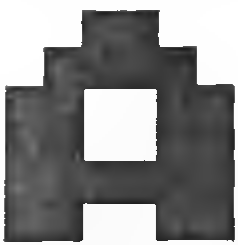
GRAPHICS by Graham

The way in which characters are produced is very easy to understand,and even more importantly very easy to modify.The code to generate the standard character set that ATARI have provided is stored in ROM,your machine knows where that character set is because it has a pointer which tells it where it can be found,this pointer is also 'shadowed' in RAM and so can be modified,The RAM character set pointer lives at 756 decimal and its usual content is 224 decimal,if you perform the following as a direct command you will see this is so:-

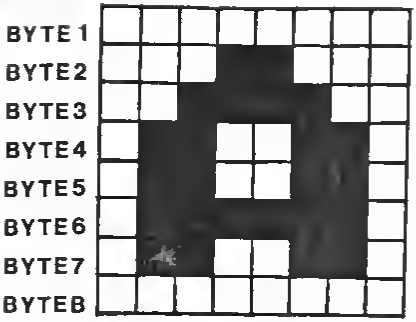
PRINT PEEK(756)

As I said before because this is a RAM location it can be modified to point at a different character set try POKE 756,152 Whoops where did all the text go,try typing some characters.....nothing,now hit system reset.....all is well again.What you just did is to tell your machine to look at a non-existant character set and so you saw nothing,the ability to be able to switch character sets under software control is very powerful.Suppose you wish to write a game and you wish to graphically represent a small man,this is not easy with the conventional block graphics that ATARI provide so why not define your own shapes ?

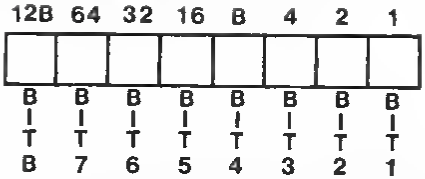
To define your own graphic blocks you must first understand how the character set is stored,if you examine the character set closely on your TV screen,you will see that each character is built of blocks in an eight by eight matrix,this is in fact one byte wide(8 bits) by eight bytes high and each bit that contains a one is illuminated,each bit that contains a zero is not,thus the letter 'A' is stored in eight bytes.



Let us look at this more closely

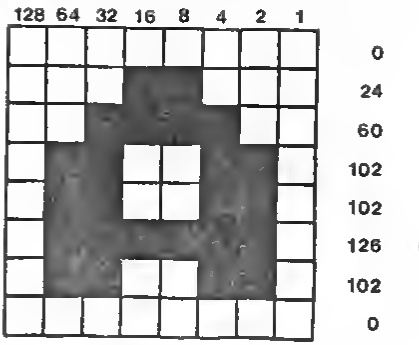


Firstly your ATARI displays byte 1 where none of the bits are set,then byte 2 where some bits are illuminated and some are not,this process continues until the eighth byte is displayed and the character is complete.How then do we tell our machine which bits to illuminate ? well it could not be easier,each byte can hold a number between 0 and 255,all we have to do is to work out which number turns on the bits we need illuminated.The byte shown below will make this clearer.



Suppose we do not wish to turn on any bits,then we simply store a '0' in that byte as in byte 1,in byte 2 we wish to turn on bits 4 and 5,these correspond to a numeric value of 8+16=24 so 24 is the number we store.In byte 3 we wish to turn on bits 3,4,5 and 6 so we store 4+8+16+32=60 and so on.

Now lets look at our letter 'A' again on this time I have added up the bytes for you.



The first step in redefining your character set is to copy the existing character set from ROM into RAM,a convienient place to start your RAM character set is 152 (this 152*256 in decimal.).The following program will copy the ROM set into Ram.

```
10 FOR A=0 to 1023:POKE 152*256+A,PEEK(224*256+A):NEXT
A
```

To prove this has worked poke the new starting address of your RAM set into the character set pointer.

```
POKE 756,152
```

Nothing dramatic happens as the new character set that you have selected is identical to the old one,but as it is now in RAM you can easily modify it.

I will show you one simple modification and then you play for yourselves.We will modify CHR$(33),which is the exclamation mark,unfortunately the character set table is stored in a different order from the CHR$ table and the exclamation mark is character two in our RAM character set.As our character set starts at 38912, you can see that the table for the exclamation mark starts eight bytes down (character zero extends from 38911 to 38919) ,if you perform the following as a direct command:-

```
PRINT PEEK(38920)
```

Your computer should print '0' ,this is because in the first byte of the eight byte table that your computer uses to store the codes for the exclamation mark,all the bits are set to zero.Lets do a simple modification,as a direct command do the following(having first run the character set copy program of course)

```
POKE 38920,255
```

this will set all the bits and should make the top line of the exclaimation mark solid white,try the poke and then PRINT CHR$(33), hey presto the system works !!

Listed on the next page is a small program that will modify all the eight bytes that hold the coding that the ATARI needs to display the exclamation mark.

```
5 FOR A=0 TO 1023:POKE 152*256+A,PEEK(57*256+A
:NEXT A:POKE 756,152
10 FOR A=0 TO 7
20 READ BYTE
30 POKE 38920+A,BYTE
40 NEXT A
50 DATA 129,90,60,126,126,60,90,129
```

After typing this program in,I suggest that you save it before running as if you have made some errors it is possible that the POKE statement will crash your computer temprorarily

Now try printing CHR$(38) and you will see your new shape,to see how I worked out the data staements refer to fig 3.That explains a simple modification,now some extra ideas for you,since you have modified your character set,whenever you use the shift 1 key your new character will appear,it is possible however to switch between character sets,try as a direct command POKE 756,224 and you will see the normal exclaimation mark return,now POKE 756,152 and the modified character will appear,thus you can switch character sets very quickly(just one poke from BASIC) and this is extremely useful for animation,in fact ATARI's Space Invaders is a classic example of character set redefinition and switching.The characters appear to flap their wings and rotate as they move across the screen,this effect is generated by rapid switching of predefined character sets held at various locations in memory.

Ne t time I will attempt to explain how to create your own graphics modes by writing custom diplay lists.This will enable you to mix various sizes of text and various resolutions of graphics together on one page.

In this issue is a listing of a program call Flan Flinger.which uses character set redefinition,it also uses some other tricky graphics ideas which I will be explaining in our following issues.In fact this program has a very funny origin,it was originally started by Chris as a very basic space attack type game,he then visited myself one weekend and we started to modify parts of it,eventually it was decided that space ship characters were getting a bit of a bore so we rehashed the program completely and ended up with Flan Flinger.The program is not brilliant,but may be useful as a example of graphics control.

```
1 DIM N(34),T(34)
2 FOR A=1 TO 34:READ N1,T:N(A)=N1:T(A)=T:NEXT A
3 DATA 162,1,128,1,121,1,108,4,162,1,128,1,121,1,108,4,162,1,128,1,121,1,108,2,1
28,2,161,2,128,2,144,4
4 DATA 128,1,128,1,144,1,162,2,162,1,128,2,108,1,108,1,108,1,121,2,121,1,128,1,1
21,1,108,2,128,2,144,2
5 DATA 144,2,162,4
9 TOP=PEEK(106)-8
10 GOSUB 7000
15 GOSUB 10000
76 GRAPHICS 2+16:POSITION 4,5:SETCOLOR 0,7,8:PRINT #6;"PRESS START"
79 POKE 755,4:FOR T=1 TO 200:NEXT T:POKE 755,2:FOR Z=1 TO 400:NEXT Z:IF PEEK(532
79)<>6 THEN 79
80 GRAPHICS 1+16
85 GOSUB 8000
92 POKE 756,TOP
97 SETCOLOR 0,15,10:SETCOLOR 4,15,2
100 REM *** DRAW BOX ***
110 FOR A=0 TO 19
120 POSITION A,0
130 PRINT #6;"■"
140 NEXT A
150 FOR A=0 TO 22
160 POSITION 19,A
170 PRINT #6;"■"
180 NEXT A
190 FOR A=19 TO 0 STEP -1
200 POSITION A,22
210 PRINT #6;"■"
220 NEXT A
230 FOR A=22 TO 0 STEP -1
240 POSITION 0,A
250 PRINT #6;"■"
260 NEXT A
265 SHIP=2:HISCORE=0
270 SCORE=0:TIME=26:SFY=18:SF=0
271 GOSUB 1000
275 INVX=INT(RND(0)*19):IF INVX<1 THEN 275
276 INVY=INT(RND(0)*19):IF INVY<2 THEN 276
300 REM *** SCORE ***
310 POSITION 5,1
315 PRINT #6;"HI-SCORE ";HISCORE
316 POSITION 1,21
320 PRINT #6;"SCORE      "
322 POSITION 1,21
323 PRINT #6;"SCORE ";SCORE
325 TIME=TIME-1
326 SOUND 0,20,10,15
330 POSITION 16,21
335 PRINT #6;"   "
336 POSITION 11,21
340 PRINT #6;"TIME ";TIME
345 SOUND 0,0,0,0
350 IF TIME=0 THEN 5000
400 REM **** JOYSTICK ****
405 FOR T=1 TO 200
```

```
407 IF SF=1 THEN 425
410 IF STICK(0)=11 THEN GOSUB 1000
420 IF STICK(0)=7 THEN GOSUB 2000
425 IF SF=1 THEN GOSUB 3000
426 IF SF=1 THEN 432
430 IF STRIG(0)=0 THEN SFY=18:SFX=SHIP+1:GOSUB 3000
432 IF T=25 THEN 325
433 GOSUB 4000
435 POKE 77,0:NEXT T
440 GOTO 410
1000 REM ** JOYSTICK MOVEMENT LEFT **
1005 SHIP=SHIP-1
1010 IF SHIP<1 THEN SHIP=1:RETURN
1020 POSITION SHIP,20
1025 PRINT #6;"YX"
1026 SOUND 0,255,10,10
1030 POSITION SHIP+2,20
1035 PRINT #6;" "
1036 SOUND 0,0,0,0
1040 RETURN
2000 REM ** JOYSTICK MOVEMENT RIGHT **
2005 SHIP=SHIP+1
2010 IF SHIP>17 THEN SHIP=17:RETURN
2015 POSITION SHIP,20
2020 PRINT #6;"YX"
2021 SOUND 0,255,10,10
2025 REM POSITION SHIP-1,20
2030 REM PRINT #6;" "
2031 SOUND 0,0,0,0
2040 RETURN
3000 REM *** SHIP FIRE ***
3030 SF=1
3040 IF SFY=INVY AND SFX=INVX THEN 5900
3050 IF STICK(0)=11 THEN GOSUB 3060
3051 IF STICK(0)=7 THEN GOSUB 3100
3053 IF SFY=1 THEN SF=0:RETURN
3054 SFY=SFY-1:IF SFY<2 THEN SF=0:RETURN
3055 POSITION SFX,SFY
3056 PRINT #6;"Q":SOUND 0,30,10,10
3057 POSITION SFX,SFY+1
3058 PRINT #6;" ":SOUND 0,0,0,0:IF SFY=2 THEN POSITION SFX,SFY:PRINT #6;" "
3059 RETURN
3060 REM ***
3062 SFX=SFX-1
3063 IF SFX<1 THEN SFX=1:RETURN
3075 POSITION SFX+1,SFY
3080 PRINT #6;" "
3090 RETURN
3100 REM **
3110 SFX=SFX+1
3120 IF SFX>18 THEN SFX=18:RETURN
3130 POSITION SFX-1,SFY
3140 PRINT #6;" "
3150 RETURN
4000 REM *** INVADER MOVEMENT ***
4010 MOV=INT(RND(0)*8)
```

```
4011 IF MOV=0 THEN 4050
4012 IF MOV=1 THEN 4100
4013 IF MOV=2 THEN 4150
4014 IF MOV=3 THEN 4200
4015 RETURN
4050 REM *** INV-UP ***
4052 INVY=INVY-1
4055 IF INVY<2 THEN INVY=2:RETURN
4060 POSITION INVX,INVY
4065 SOUND 0,80,10,5
4066 IF SF=1 AND INVY<SFY THEN PRINT #6;"W":GOTO 4075
4070 PRINT #6;"Z"
4075 POSITION INVX,INVY+1
4077 SOUND 0,0,0,0
4080 PRINT #6;" "
4090 RETURN
4100 REM *** INV-DOWN ***
4105 INVY=INVY+1
4106 IF INVY>18 THEN INVY=18:RETURN
4110 POSITION INVX,INVY
4112 SOUND 0,90,10,5
4113 IF SF=1 AND INVY<SFY THEN PRINT #6;"W":GOTO 4120
4115 PRINT #6;"Z"
4120 POSITION INVX,INVY-1
4125 PRINT #6;" "
4127 SOUND 0,0,0,0
4130 RETURN
4150 REM *** INV-LEFT ***
4155 INVX=INVX-1
4156 IF INVX<1 THEN INVX=1:RETURN
4157 POSITION INVX,INVY
4158 SOUND 0,100,10,5
4159 IF SF=1 AND INVY<SFY THEN PRINT #6;"W":GOTO 4165
4160 PRINT #6;"Z"
4165 POSITION INVX+1,INVY
4167 SOUND 0,0,0,0
4170 PRINT #6;" "
4180 RETURN
4200 REM *** INV-RIGHT ***
4205 INVX=INVX+1
4206 IF INVX>18 THEN INVX=18:RETURN
4210 POSITION INVX,INVY
4212 SOUND 0,110,10,5
4213 IF SF=1 AND INVY<SFY THEN PRINT #6;"W":GOTO 4220
4215 PRINT #6;"Z"
4220 POSITION INVX-1,INVY
4222 SOUND 0,0,0,0
4225 PRINT #6;" "
4230 RETURN
5000 REM *** END ***
5001 POSITION INVX,INVY:PRINT #6;" "
5002 POSITION SFX,SFY:PRINT #6;" "
5003 IF SCORE>HISCORE THEN HISCORE=SCORE
5005 POSITION 14,1
5006 PRINT #6;HISCORE
5010 POSITION 5,3
```

```
5020 PRINT #6;"GAME OVER"
5030 POSITION 5,5
5040 PRINT #6;"PRESS START"
5045 SOUND 0,250,10,10:SOUND 1,255,10,10:FOR I=1 TO 150:NEXT I:SOUND 0,0,0,0:SOU
ND 1,0,0,0
5050 POKE 755,4:FOR T=1 TO 20:NEXT T:POKE 755,2:FOR T=1 TO 30:NEXT T:IF PEEK(532
79)<>6 THEN 5050
5060 POSITION 5,3
5070 PRINT #6;"          "
5080 POSITION 5,5
5090 PRINT #6;"            "
5095 POSITION SHIP,20
5096 PRINT #6;" "
5100 GOTO 270
5900 SFX=INVX:SFY=INVY:POSITION SFX,SFY
5910 PRINT #6;" "
6000 GOSUB 9000
6010 SCORE=SCORE+INVY
6040 POSITION SFX,SFY
6050 PRINT #6;"x":FOR T=1 TO 40:NEXT T
6060 POSITION INVX,INVY
6070 PRINT #6;" "
6080 INVX=INT(RND(0)*19):IF INVX<1 THEN 6080
6090 INVY=INT(RND(0)*19):IF INVY<2 THEN 6090
6100 SOUND 0,0,0,0:SF=0:TIME=TIME+Z:GOTO 300
7000 GOSUB 12000:FOR A=0 TO 1023
7004 IF A>989 THEN GOSUB 14000
7005 POKE TOP*256+A,PEEK(57344+A):NEXT A
7010 FOR Z1=0 TO 7:READ BYTE:POKE (TOP*256)+448+Z1,BYTE:NEXT Z1
7020 DATA 28,55,94,119,191,201,255,0
7030 FOR Z1=0 TO 7:READ BYTE:POKE (TOP*256)+464+Z1,BYTE:NEXT Z1
7040 DATA 126,255,153,255,219,195,255,126
7050 FOR Z1=0 TO 7:READ BYTE:POKE (TOP*256)+392+Z1,BYTE:NEXT Z1
7060 DATA 4,30,55,94,119,62,0,0
7070 FOR Z1=0 TO 7:READ BYTE:POKE (TOP*256)+456+Z1,BYTE:NEXT Z1
7080 DATA 0,0,0,255,255,255,255,255
7090 FOR Z1=0 TO 7:READ BYTE:POKE (TOP*256)+24+Z1,BYTE:NEXT Z1
7100 DATA 255,129,189,165,165,189,129,255
7110 FOR Z1=0 TO 7:READ BYTE:POKE (TOP*256)+440+Z1,BYTE:NEXT Z1
7120 DATA 126,255,153,189,255,195,219,126
7130 RETURN
8000 REM ****OLI****
8010 OL=PEEK(560)+PEEK(561)*256:POKE OL+25,7
8020 POKE OL+24,134:FOR I=0 TO 19:READ BYTE:POKE 1536+I,BYTE:NEXT I
8030 DATA 72,138,72,169,21,162,37,141,10,212,141,27,208,141,26,208,104,170,104,6
4
8040 POKE 512,0:POKE 513,6:POKE 54286,192
8050 RETURN
9000 REM *****SPLAT*****
9005 E=0
9006 POSITION SFX,SFY:? #6;"x"
9007 POSITION 7,9:? #6;"SPLAT"
9010 SOUND 0,240-E,8,8
9015 SOUND 1,220-E-4,8,8
9017 POSITION 7,9:? #6;"      "
9020 E=E+8:IF E=96 THEN 9040
```

```
9030 GOTO 9007
9040 SOUND 1,0,0,0:SOUND 0,0,0,0:FOR T=1 TO 10:NEXT T:SOUND 0,200,8,12:FOR T=1 T
O 20:NEXT T
9045 POSITION 7,9:? #6;"      "
9050 SOUND 0,0,0,0:RETURN
10000 REM ****INTRO*****
10005 COL0=708:COL2=710:SO0=(13*16+2)*256+1:SO1=SO0+2:SO2=SO1+2:SO3=SO2+2
10010 GRAPHICS 2+16
10020 PRINT #6;"**********************"
10030 FOR I=1 TO 10:POSITION 0,I:PRINT #6;"*                    *";
10040 NEXT I
10045 PRINT #6;"**********************";
10050 POSITION 4,2:PRINT #6;"THE  PHANTOM"
10055 POSITION 4,4:PRINT #6;"FLAN FLINGER"
10070 POSITION 6,6:PRINT #6;"MAPSOFT"
10075 POSITION 6,8:PRINT #6;"1981(c)"
10077 SOUND 0,40,0,0:SOUND 1,35,0,0:SOUND 2,31,0,0:SOUND 3,45,0,0
10080 POKE 764,255:FOR H=1 TO 10 STEP 5
10090 FOR I=4 TO 15
10092 CR=I+5:IF CR>15 THEN CR=16-I
10094 SETCOLOR 1,CR,8:SETCOLOR 3,15-CR,8
10095 FOR L=0 TO 15 STEP 2
10100 C=I*16+L:POKE COL0,C:POKE COL2,255-C
10105 J=176-L:K=48-L
10107 POKE SO0,J:POKE SO1,J:POKE SO2,J:POKE SO3,J
10108 NEXT L
10109 IF PEEK(764)=39 THEN 10127
10110 NEXT I
10120 NEXT H
10125 SETCOLOR 0,2,8:SETCOLOR 2,9,4
10126 FOR O=1 TO 40:NEXT O
10127 SOUND 0,0,0,0:SOUND 1,0,0,0:SOUND 2,0,0,0:SOUND 3,0,0,0:RETURN
12000 GRAPHICS 1+16:SETCOLOR 0,15,2:SETCOLOR 4,15,6
12010 ? #6;"USE STICK 0 TO MOVE"
12020 ? #6;"YOUR FLAN FLINGING"
12030 ? #6;"HAND BACK AND FORTH"
12031 ? #6;" "
12032 ? #6;"PUSH THE REO BUTTON"
12033 ? #6;"TO FLING YOUR FLAN"
12034 ? #6;" "
12040 ? #6;"WHEN YOU HAVE FLUNG"
12050 ? #6;"YOUR FLAN YOU MAY"
12060 ? #6;"STEER IT."
12061 ? #6;" "
12080 ? #6;"POINTS ARE AWARDED"
12090 ? #6;"WHEN YOU STRIKE !!"
12091 ? #6;" "
12100 ? #6;"THE CLOSER YOUR"
12110 ? #6;"VICTIM THE HIGHER"
12120 ? #6;"YOUR POINTS."
12130 ? #6;" "
12140 ? #6;"WATCH YOUR TIMER"
12150 ? #6;"YOU WILL BE AWARDED"
12160 ? #6;"EXTRA TIME FOR EACH"
12170 ? #6;"VICTIM HIT."
12180 RETURN
14000 REM ****PLAY IT***
14010 Z2=22+1
14012 IF 22>34 THEN SOUND 0,0,0,0:SOUND 1,0,0,0:Z2=0:RETURN
14014 T=T(22)*5:FOR J=1 TO T
14020 SOUND 0,N(22),10,8:SOUND 1,N(22)+2,10,8
14030 NEXT J
14040 RETURN
```

WHAT ATARI FORGOT TO TELL THEIR 400 OWNERS.

When the ATARI computers were first made,a special television interface chip was designed,this chip was called C.T.I.A this chip is what gives the ATARI computers their great graphics flexibility,however during the re-designing of the machine for the PAL television standard a new graphics chip became available called G.T.I.A and all your machines have a G.T.I.A fitted,but your documentation does not tell you how to use it !

C.T.I.A and G T.I.A are designed to work with a chip called ANTIC which provides data that enables them to function.C.T.I.A will interpret data for six text modes and eight graphic modes.C.T.I.A can handle a total of sixteen modes,I will concentrate on two of the new ones that can be accessed from BASIC ,these are graphics modes 9 and 11.

Each has a resolution of 80 by 192 pixels,not very exciting you say,but,and this is the crunch ,in graphics mode 9 you can use 16 luminances of the same hue and graphics mode 11 has 16 hues of the same luminance.Now you have a lot of color to play with !

Lets look at the nuts and bolts,first mode 9.

You must first select your background color,this is done using the SETCOLOR 4 command,its format is:-

SETCOLOR 4,Hue value,0

Hue value can be any number between 0 and 15 of course,the luminance number must be set to 0,to change luminance you can use the COLOR statement,try:-

```
5 GR.9:SETCOLOR 4,15,0
10 FOR A=0 TO 15
20 COLOR A
30 PLOT A,0
40 DRAWTO A,191
50 NEXT A
60 GOTO 60
```

Line 5 selects graphics 9,line 10 sets up the loop,line 20 changes the luminance,line 30 plots a single point,line 40 performs a drawto from the last plotted point,line 50 is the end of the loop.line 60 is necessary because in graphics 9 and 11 if your program is actually completed GTIA will return you automatically to graphics 0.

For a better example see the program on the following page called BOX.V1.the above listing has been left deliberately simple.Some other examples for you to play with that use graphics 9 are PAT1 and BOX.V2,these can be found at the end of this article.

On to graphics mode 11, which is very similar to 9 except that this time you can control the hue instead of the luminance,you use the same SETCOLOR statement ie

```
SETCOLOR 4,0,Luminance value
```

Where the second parameter must be 0 and the luminance value can be any number between 0 and 15,except of course,as usual,the first bit is not used and so only even numbers will make changes (if you didn't know this already then perhaps a re-read of the SETCOLOR summery in the manual in order),try :-

```
10 GR.11
20 SETCOLOR 4,0,6
30 FOR A=0 TO 15
40 COLOR A
50 PLOT A,0
60 DRAWTO A,151
70 NEXT A
80 GOTO 80
```

Line 10 selects graphics 11,line 20 sets the luminance of any selected color to 6,line 30 starts the loop to change colors,line 40 changes the color,lines 50 and 60 include the plot and drawing of the line,line 70 completes the loop and line 80 makes sure that your machine stays in a loop otherwise it will automatically drop out to graphics 0.

Again the above example is perhaps over simple and you might care to play around with BOX.V2 and again I have included some more material for the experimenter in the form of PAT2 and CIRCLE1 (the circle plot routine may be of interest),these programs were developed by Chris to demonstrate basic ideas and no prizes will be awarded for improving the code as they have been presented just as they were written 'from the top down'!!!!

```
1 REM **********************BOX.V1********************
2 GRAPHICS 9
5 X1=0:X2=79:Y1=0:C=15:Y2=191:R1=1
10 SETCOLOR 4,7,0
25 COLOR C
30 X1=X1+0.42:X2=X2-0.42:Y1=Y1+R1:C=C-0.15:Y2=Y2-R1
35 IF Y1>96 THEN 70
40 PLOT X1,Y1:DRAWTO X2,Y1
45 PLOT X1,Y2:DRAWTO X2,Y2
46 PLOT X1,Y1:DRAWTO X1,Y2
47 PLOT X2,Y1:DRAWTO X2,Y2
50 GOTO 25
70 GOTO 70
```

```
5 REM ********************PAT1**********************
10 C=0:G=13
20 GRAPHICS 9
30 FOR A=0 TO 79 STEP 2:PLOT A,0:DRAWTO 79-A,191
31 SOUND 0,A+30,10,5:SOUND 1,A+33,10,5:SOUND 2,A+36,10,5:SOUND 3,A+39,10,5
32 C=C+1:IF C=16 THEN C=1
33 COLOR C
34 NEXT A
40 FOR A=191 TO 0 STEP -4:PLOT 0,A:DRAWTO 79,191-A
41 SOUND 0,A+30,10,5:SOUND 1,A+33,10,5:SOUND 2,A+36,10,5:SOUND 3,A+39,10,5
42 C=C+1:IF C=16 THEN C=2
43 COLOR C
44 NEXT A
100 SOUND 0,0,0,0:SOUND 1,0,0,0:SOUND 2,0,0,0:SOUND 3,0,0,0:GOTO 100
```

```
1 REM **********************BOX.V2********************
2 GRAPHICS 9:R=0.15:S=7:V=96
5 X1=0:X2=79:Y1=0:C=15:Y2=191:R1=1
10 SETCOLOR 4,S,0
30 X1=X1+0.42:X2=X2-0.42:Y1=Y1+R1:C=C-R:Y2=Y2-R1
35 IF Y1>V THEN 70
36 IF C<0 THEN C=15
38 COLOR C
40 PLOT X1,Y1:DRAWTO X2,Y1
45 PLOT X1,Y2:DRAWTO X2,Y2
46 PLOT X1,Y1:DRAWTO X1,Y2
47 PLOT X2,Y1:DRAWTO X2,Y2
50 GOTO 30
70 R=RND(0)*1:S=INT(RND(1)*16):GOTO 5
```

```
1 REM ********************BOX,V3**********************
2 GRAPHICS 11:R=0.15:S=7:V=96
5 X1=0:X2=79:Y1=0:C=15:Y2=191:R1=1
30 X1=X1+0.65:X2=X2-0.65:Y1=Y1+R1:C=C-R:Y2=Y2-R1
35 IF Y1>V THEN 70
36 IF C<0 THEN C=15
38 COLOR C
40 PLOT X1,Y1:DRAWTO X2,Y1
45 PLOT X1,Y2:DRAWTO X2,Y2
46 PLOT X1,Y1:DRAWTO X1,Y2
47 PLOT X2,Y1:DRAWTO X2,Y2
50 GOTO 30
70 R=RND(0)*1:GOTO 5
```

```
5 REM ************************PAT2************************
10 C=1:G=13
20 GRAPHICS 11:SETCOLOR 4,0,6
30 FOR A=0 TO 79 STEP 2:PLOT A,0:DRAWTO 79-A,191
31 SOUND 0,A+30,10,5:SOUND 1,A+33,10,5:SOUND 2,A+36,10,5:SOUND 3,A+39,10,5
32 C=C+0.2:IF C>15 THEN C=1
33 COLOR C
34 NEXT A
40 FOR A=191 TO 0 STEP -4:PLOT 0,A:DRAWTO 79,191-A
41 SOUND 0,A+30,10,5:SOUND 1,A+33,10,5:SOUND 2,A+36,10,5:SOUND 3,A+39,10,5
42 C=C+0.2:IF C>15 THEN C=1
43 COLOR C
44 NEXT A
100 GOTO 30
```

```
1 REM ********************CIRCLE1*********************
5 DEG :SOUND 0,255,10,2:SOUND 1,254,10,2:SOUND 2,253,10,2:SOUND 3,252,10,2
10 GRAPHICS 10
20 X=22:Y=96:A=1:C=4:GOTO 80
21 FOR Q=1 TO 138
22 C=C+R:IF C>15 THEN C=4
25 COLOR C
30 PLOT 45,90
40 DRAWTO X,Y
45 A=A+2.6
50 X=X+SIN(A)
55 Y=Y+COS(A)*4
56 POKE 77,0
60 NEXT Q
80 R=RND(1)*9:P=RND(1)*16:SETCOLOR 4,P,10:GOTO 21
```

Graham's letter page.

A letter from Derek Lees raises a number of points,apparently Derek purchased his 400 from a shop which is part of a larger chain who have decided not to offer a 32K upgrade because of Ingersoll's refusal to support machines that have been upgraded,apparently this particular shop also states that the 32K board causes overheating in the 400.

I offer this advice to Derek,firstly it is true that Ingersoll have decided not to service upgraded machines and I find this quite understandable as they have no guarantee as to the technical ability of the people doing the upgrade.With regard to the over heating rumour,I too have been told this in a telephone conversation with the same company,however I can tell you with great confidence(being involved with the company who has shipped the most 32K machines) that I have been tested all the 32K upgrades and there is only one that overheats and then it will not overheat in an 800,if you decide to have a 32K upgrade performed by another retailer then check first that he is prepared to offer you a guarantee on the upgrade,I do and can see no reason why anyone should not.

Next is a letter from Jon Williams who firstly offers me compliments and then gets down to business,Jon writes mainly in assembler using the and I quote 'super slow Atari cartridge ',Jon wants to know why the trace and single step modes either break or crash when they reach a CMY opcode.

Thanks for the compliments Jon and you will be pleased to know that a macro assembler editor will be available for your machine before the middle of the year,with regard to your problem,I don't have an answer so this is the first chance for someone out there to help,please write to me at the newsletter address and I will forward your letters to Jon.

Mr D Terry from Suffolk writes to ask if spares for Joysticks are available,the answer to that is no,however I appreciate your problems with those white nylon bits,they seem to have got really bad of late.Most Atari dealers should replace broken joysticks within a reasonable period of time.

The following letter was recieved from Jonothen Senders
24,Greenacres,
Birdhem
Chichester
w.Sussex

After reading e recent article in December BYTE magezine,I wes eble to connect a simple Cadmium sulphide cell and thermistor to pins 5 and 7 of the front port of my ATARI.Then using the instruction PADDLE(0), I could use the ATARI to detect light chenges or temperature chenges.

I would like to use my ATARI to operate a switching circuit,has any club member developed a simple circuit for switching which can be connected to either the front ports or the interfece box ?

In addition I have successfully used my ATARI to sign onto severel of the U.K networks and Bulletin boards.Heve eny other members tried to do this and if so would they like to exchange experiences/progrems.If so contact me directly or leeve a message on FORUM 80 or CBBS Bulletin Boards.

Jonathan and I have met and both have a very deep interest in the ATARI so if enyone can help him I can assure them that he will be able to offer a lot of experience,the idea of the switch box is a good one and I would be please to publish any designs that enyone comes up with,also I wish to publish some terminal emulators end communicetions software so if anyone has anything et all then please forward it to myself or Jonethen.

Finally:- MR J Baker
Home close
Sheep street
Winslow
Buckingham
MK18 3Hr

Mr Baker is attempting to use a conventional cessette recorder with his ATARI insteed of the 410,he is experiencing problems and would appreciate assistance from anyone with any ideas,please contact Mr Baker direct at the above eddress.

THATS ALL THIS TIME PLEASE ADDRESS ALL ANSWERS OR QUESTIONS TO GRAHAM AT THE GROUP ADDRESS OR ANSWERS DIRECT IF THE MEMBERS ADDRESS IS GIVEN,WOULD MEMBERS RECIEVING ANSWERS DIRECT THAT WORK PLEASE FORWARD THEM TO GRAHAM FOR PUBLICATION.

MORSE TUTOR PROGRAM

by Chris

Morse code was developed for use in electric telegraphy in 1838 by Samuel F.B. Morse.

The earliest transatlantic radio signal was recieved by Marconi at Poldhu, Cornwall, from a 10 KW transmitter at Newfoundland,Canada,on the 12th December 1901.The highest speed ever recorded for recieving morse code is 75.2 words per minute and 175 w.p.m for sending by hand.

Morse transmissions can be recieved on various short wave bands,emateur radio end shipping signals etc.These morse signals are generally between 8 end 15 w.p.m,some use greater speeds however.

This program generates rendom morse chersacters,letters and/or numbers at 5 to 30 w.p.m.You can control the delay between morse characters and the tone(pitch).

C.W signal sets the strength of the morse recieved,5/9 is a strong signal and 5/1 is very weak.As the morse gets weaker a larger amount of interfering noise is introduced.

Tutor length sets the number of morse characters generated from 10 to 430,but you can stop at anytime by pressing the option key.

After all the requested number of charecters have been sent,an end bleep will sound and you may then read from the screen to check your accuracy.

You may then use the start key to repeat the exercise or press select key for the setup menu

When the program is running use option/select/start keys to control the program and here is a brief description of the working program:-

1000 to 1540 read random line of data
1550 end 1560 CW tone and QRM sound
1600 print morse characters and delay/length routine
1700 to 1760 end of morse and setcolor to read screen
2000 to 2035 morse data,dit,dah and space.last number is character string value
2500 and 2510 setup routine for tone and speed etc.
3000 to 3140 print graphics page 2 menu/setup
3150 to 3525 select/start routines tone speed etc.
4000 to 4230 print graphics page 1 the morse code
5000 start morse sounds routine.

The complete program is now listed.

```
20 GOSUB 2500
1000 A=INT(RND(1)*TYPE)
1005 IF TYPE>10 AND A<10 AND MIXED=0 THEN 1000
1010 IF PEEK(53279)=3 THEN 1700
1040 RESTORE A+2000
1500 READ B
1510 IF B=1 THEN C=TONE:X=1
1520 IF B=2 THEN C=TONE:X=3
1530 IF B=3 THEN C=0:X=1
1540 IF B>3 THEN C=0:X=3
1550 SOUND 0,C,10,SIGNAL:SOUND 1,17,8,ORM:SOUND 2,1,4,ORM:SOUND 3,18,10,ORM
1560 FOR T=0 TO X*RATE:NEXT T
1570 IF B>3 THEN 1600
1580 GOTO 1500
1600 POKE 77,0:PRINT CHR$(B);":";:FOR T=0 TO DELAY*360:NEXT T:LENGTH=LENGTH+1:IF
 LENGTH=TIME THEN 1700
1610 GOTO 1000
1700 SOUND 1,0,0,0:SOUND 2,0,0,0:SOUND 3,0,0,0:SETCOLOR 1,0,10:SETCOLOR 2,7,0
1710 FOR T=0 TO 10:SOUND 0,70,10,5:SOUND 1,71,10,5:SOUND 2,72,10,5:SOUND 3,73,10
,5
1720 FOR C=0 TO 30:NEXT C:SOUND 0,0,0,0:SOUND 1,0,0,0:SOUND 2,0,0,0:SOUND 3,0,0,
0
1730 FOR C=0 TO 30:NEXT C:NEXT T
1740 IF PEEK(53279)=5 THEN GOSUB 3000
1745 IF PEEK(53279)=6 THEN GOSUB 5000
1750 IF LENGTH=0 THEN 1000
1760 GOTO 1740
2000 DATA 2,3,2,3,2,3,2,3,2,48
2001 DATA 1,3,2,3,2,3,2,3,2,49
2002 DATA 1,3,1,3,2,3,2,3,2,50
2003 DATA 1,3,1,3,1,3,2,3,2,51
2004 DATA 1,3,1,3,1,3,1,3,2,52
2005 DATA 1,3,1,3,1,3,1,3,1,53
2006 DATA 2,3,1,3,1,3,1,3,1,54
2007 DATA 2,3,2,3,1,3,1,3,1,55
2008 DATA 2,3,2,3,2,3,1,3,1,56
2009 DATA 2,3,2,3,2,3,2,3,1,57
2010 DATA 1,3,2,65
2011 DATA 2,3,1,3,1,3,1,66
2012 DATA 2,3,1,3,2,3,1,67
2013 DATA 2,3,1,3,1,68
2014 DATA 1,69
2015 DATA 1,3,1,3,2,3,1,70
2016 DATA 2,3,2,3,1,71
2017 DATA 1,3,1,3,1,3,1,72
2018 DATA 1,3,1,73
2019 DATA 1,3,2,3,2,3,2,74
2020 DATA 2,3,1,3,2,75
2021 DATA 1,3,2,3,1,3,1,76
2022 DATA 2,3,2,77
2023 DATA 2,3,1,78
2024 DATA 2,3,2,3,2,79
2025 DATA 1,3,2,3,2,3,1,80
2026 DATA 2,3,2,3,1,3,2,81
2027 DATA 1,3,2,3,1,82
2028 DATA 1,3,1,3,1,83
```

```
2029 DATA 2,84
2030 DATA 1,3,1,3,2,85
2031 DATA 1,3,1,3,1,3,2,86
2032 DATA 1,3,2,3,2,87
2033 DATA 2,3,1,3,1,3,2,88
2034 DATA 2,3,1,3,2,3,2,89
2035 DATA 2,3,2,3,1,3,1,90
2500 SIGNAL=10:ORM=0:DELAY=0:TONE=10:SPEED=12:TYPE=36:TIME=30:CODE=0:CAL=100:RX=
9:CWL=1:CWN=1
2510 RATE=INT(CAL/SPEED*2)
3000 REM ******MENU*******
3010 GRAPHICS 2+16:SETCOLOR 4,7,1:SETCOLOR 0,4,8
3020 FOR D=0 TO 19:POSITION D,0:PRINT #6;"*":NEXT D
3030 FOR D=0 TO 19:POSITION D,10:PRINT #6;"*":NEXT D
3040 FOR D=0 TO 9:POSITION 0,D:PRINT #6;"*":NEXT D
3050 FOR D=0 TO 9:POSITION 19,D:PRINT #6;"*":NEXT D
3060 POSITION 1,1:PRINT #6;"THE CODE,,,,,,, ?"
3062 IF CODE=1 THEN POSITION 14,1:PRINT #6;"YES"
3065 IF CODE=0 THEN POSITION 14,1:PRINT #6;"NO."
3070 POSITION 1,2:PRINT #6;"SPEED(WPM)...";SPEED;". ?"
3072 IF SPEED<10 THEN POSITION 15,2:PRINT #6;".. ?"
3080 POSITION 1,3:PRINT #6;"DELAY(SEC)...";DELAY;" ?"
3085 IF DELAY=INT(DELAY) THEN POSITION 15,3:PRINT #6;".0 ?"
3090 POSITION 1,4:PRINT #6;"C.W.TONE....";TONE;". ?"
3100 POSITION 1,5:PRINT #6;"C.W.SIGNAL..5/";RX;" ?"
3110 POSITION 1,6:PRINT #6;"C.W.LETTERS.... ?"
3112 IF CWL=1 THEN POSITION 14,6:PRINT #6;"YES"
3115 IF CWL=0 THEN POSITION 14,6:PRINT #6;"NO."
3120 POSITION 1,7:PRINT #6;"C.W.NUMBERS..... ?"
3122 IF CWN=1 THEN POSITION 14,7:PRINT #6;"YES"
3125 IF CWN=0 THEN POSITION 14,7:PRINT #6;"NO."
3130 POSITION 1,8:PRINT #6;"TUTOR LENGTH.";TIME
3132 IF TIME<100 THEN POSITION 16,8:PRINT #6;". ?"
3135 IF TIME>99 THEN POSITION 17,8:PRINT #6;" ?"
3140 POSITION 1,9:PRINT #6;"USE SELECT/START ?"
3150 REM *******SELECT/START**********
3155 POSITION 17,1:PRINT #6;"<"
3160 IF PEEK(53279)=5 AND CODE=1 THEN CODE=0:POSITION 14,1:PRINT #6;"NO.":SOUND
0,150,10,5:FOR I=0 TO 50:NEXT I
3165 IF PEEK(53279)=5 AND CODE=0 THEN CODE=1:POSITION 14,1:PRINT #6;"YES":SOUND
0,20,10,5:FOR I=0 TO 30:NEXT I
3166 SOUND 0,0,0,0
3170 IF PEEK(53279)=6 AND CODE=1 THEN 4000
3175 IF PEEK(53279)=6 AND CODE=0 THEN POSITION 17,1:PRINT #6;" ":SOUND 0,30,10,5
:FOR I=0 TO 30:NEXT I:GOTO 3185
3180 GOTO 3160
3185 SOUND 0,0,0,0:POSITION 17,2:PRINT #6;"<"
3190 IF PEEK(53279)=5 THEN SOUND 0,20,10,5:GOSUB 3210
3195 IF PEEK(53279)=6 THEN Z=Z+1:IF Z>10 THEN 3220
3200 GOTO 3190
3210 SPEED=SPEED+1:IF SPEED>30 THEN SPEED=5
3211 RATE=INT(CAL/SPEED*2)
3215 POSITION 15,2:PRINT #6;".":POSITION 14,2:PRINT #6;SPEED:FOR I=0 TO 30:NEXT
I:SOUND 0,0,0,0:RETURN
3220 Z=0:SOUND 0,30,10,5:POSITION 17,2:PRINT #6;" ":FOR I=0 TO 30:NEXT I:SOUND 0
,0,0,0
```

```
3225 POSITION 17,3:PRINT #6;":"
3230 IF PEEK(53279)=5 THEN DELAY=DELAY+0.1:SOUND 0,30,10,5:GOSUB 3234
3232 IF PEEK(53279)=6 THEN Z=Z+1:IF Z>10 THEN SOUND 0,30,10,5:GOTO 3240
3233 GOTO 3230
3234 IF DELAY>4 THEN DELAY=0
3235 POSITION 16,3:PRINT #6;"0":POSITION 14,3:PRINT #6;DELAY:FOR I=0 TO 30:NEXT
I:SOUND 0,0,0,0:RETURN
3240 Z=0:POSITION 17,3:PRINT #6;" ":POSITION 17,4:PRINT #6;"<":FOR I=0 TO 30:NE
T I:SOUND 0,0,0,0
3245 IF PEEK(53279)=5 THEN TONE=TONE+1:GOSUB 3260
3250 IF PEEK(53279)=6 THEN Z=Z+1:IF Z>10 THEN SOUND 0,30,10,5:GOTO 3270
3255 GOTO 3245
3260 IF TONE>99 THEN TONE=10
3265 POSITION 14,4:PRINT #6;TONE:SOUND 0,TONE,10,10:FOR I=0 TO 10:NEXT I:SOUND
,0,0,0:RETURN
3270 Z=0:POSITION 17,4:PRINT #6;" ":POSITION 17,5:PRINT #6;"<":FOR I=0 TO 30:NE
T I:SOUND 0,0,0,0
3275 IF PEEK(53279)=5 THEN RX=RX-1:GOSUB 3285
3280 IF PEEK(53279)=6 THEN Z=Z+1:IF Z>10 THEN SOUND 0,30,10,5:GOTO 3350
3284 GOTO 3275
3285 IF RX<1 THEN RX=9
3290 POSITION 16,5:PRINT #6;RX:SOUND 0,30,10,5:FOR I=0 TO 30:NEXT I:SOUND 0,0,0,
0
3295 IF RX=9 THEN SIGNAL=10:ORM=0
3300 IF RX=8 THEN SIGNAL=10:ORM=1
3305 IF RX=7 THEN SIGNAL=9:ORM=2
3310 IF RX=6 THEN SIGNAL=9:ORM=3
3315 IF RX=5 THEN SIGNAL=8:ORM=4
3320 IF RX=4 THEN SIGNAL=7:ORM=5
3325 IF RX=3 THEN SIGNAL=5:ORM=6
3330 IF RX=2 THEN SIGNAL=4:ORM=7
3340 IF RX=1 THEN SIGNAL=3:ORM=7
3345 RETURN
3350 Z=0:POSITION 17,5:PRINT #6;" ":POSITION 17,6:PRINT #6;"<":FOR I=0 TO 30:NEX
T I:SOUND 0,0,0,0
3355 IF PEEK(53279)=5 AND CWL=1 THEN SOUND 0,150,10,5:GOSUB 3330
3360 IF PEEK(53279)=5 AND CWL=0 THEN SOUND 0,30,10,5:GOSUB 3390
3365 IF PEEK(53279)=6 THEN Z=Z+1:IF Z>10 THEN SOUND 0,30,10,5:GOTO 3400
3370 GOTO 3355
3380 POSITION 14,6:PRINT #6;"NO.":CWL=0:TYPE=10:FOR I=0 TO 30:NEXT I:SOUND 0,0,0
,0:RETURN
3390 POSITION 14,6:PRINT #6;"YES":CWL=1:TYPE=36:FOR I=0 TO 30:NEXT I:SOUND 0,0,0
,0:RETURN
3400 Z=0:POSITION 17,6:PRINT #6;" ":POSITION 17,7:PRINT #6;"<":FOR I=0 TO 30:NEX
T I:SOUND 0,0,0,0
3405 IF PEEK(53279)=5 AND CWN=1 THEN SOUND 0,150,10,5:GOSUB 3430
3410 IF PEEK(53279)=5 AND CWN=0 THEN SOUND 0,30,10,5:GOSUB 3440
3415 IF PEEK(53279)=6 THEN Z=Z+1:IF Z>10 THEN SOUND 0,30,10,5:GOTO 3450
3420 GOTO 3405
3430 POSITION 14,7:PRINT #6;"NO.":CWN=0:FOR I=0 TO 30:NEXT I:SOUND 0,0,0,0:RETUR
N
3440 POSITION 14,7:PRINT #6;"YES":CWN=1:FOR I=0 TO 30:NEXT I:SOUND 0,0,0,0:RETUR
N
3450 IF CWL=0 AND CWN=0 THEN SOUND 0,70,10,5:FOR I=0 TO 40:NEXT I:SOUND 0,0,0,0:
GOTO 3470
3455 IF CWL=1 AND CWN=1 THEN MIXED=1
```

```
3456 IF CHL<CHN OR CHL>CHN THEN MIXED=0
3460 GOTO 3480
3470 POSITION 17,7:PRINT #6;" ":GOTO 3050
3480 Z=0:POSITION 17,7:PRINT #6;" ":POSITION 17,8:PRINT #6;"<":FOR I=0 TO 30:NEX
T I:SOUND 0,0,0,0
3485 IF PEEK(53279)=5 THEN TIME=TIME+10:SOUND 0,30,10,5:GOSUB 3500
3490 IF PEEK(53279)=6 THEN Z=Z+1:IF Z>10 THEN SOUND 0,30,10,5:GOTO 3510
3495 GOTO 3485
3500 IF TIME>430 THEN TIME=10:POSITION 16,8:PRINT #6;"."
3505 POSITION 14,8:PRINT #6;TIME:FOR I=0 TO 15:NEXT I:SOUND 0,0,0,0:RETURN
3510 Z=0:POSITION 17,8:PRINT #6;" ":POSITION 17,9:PRINT #6;"<":FOR I=0 TO 30:NEX
T I:SOUND 0,0,0,0
3515 IF PEEK(53279)=5 THEN SOUND 0,70,10,5:FOR I=0 TO 30:NEXT I:GOTO 3000
3520 IF PEEK(53279)=6 THEN SOUND 0,30,10,5:FOR I=0 TO 30:NEXT I:SOUND 0,0,0,0:GO
TO 5000
3525 GOTO 3515
4000 REM ****THE MORSE CODE**********
4005 GRAPHICS 1+16:SETCOLOR 4,4,1:SETCOLOR 0,13,10
4010 FOR D=0 TO 19:POSITION D,0:PRINT #6;"*":NEXT D
4015 FOR D=0 TO 19:POSITION D,22:PRINT #6;"*":NEXT D
4020 FOR D=0 TO 22:POSITION 0,D:PRINT #6;"*":NEXT D
4025 FOR D=0 TO 22:POSITION 19,D:PRINT #6;"*":NEXT D
4030 POSITION 1,1:PRINT #6;" A *-      V ***-"
4035 POSITION 1,2:PRINT #6;" B -***    W *--"
4040 POSITION 1,3:PRINT #6;" C -*-*    X -**-"
4045 POSITION 1,4:PRINT #6;" D -**     Y -*--"
4050 POSITION 1,5:PRINT #6;" E *       Z --**"
4055 POSITION 1,6:PRINT #6;" F **-*"
4060 POSITION 1,7:PRINT #6;" G --*     1 *----"
4065 POSITION 1,8:PRINT #6;" H ****    2 **---"
4070 POSITION 1,9:PRINT #6;" I **      3 ***--"
4075 POSITION 1,10:PRINT #6;" J *---    4 ****-"
4080 POSITION 1,11:PRINT #6;" K -*-     5 *****"
4085 POSITION 1,12:PRINT #6;" L *-**    6 -****"
4090 POSITION 1,13:PRINT #6;" M --      7 --***"
4095 POSITION 1,14:PRINT #6;" N -*      8 ---**"
4100 POSITION 1,15:PRINT #6;" O ---     9 ----*"
4105 POSITION 1,16:PRINT #6;" P *--*    0 -----"
4110 POSITION 1,17:PRINT #6;" Q --*-"
4115 POSITION 1,18:PRINT #6;" R *-*"
4120 POSITION 1,19:PRINT #6;" S ***     * = DIT"
4125 POSITION 1,20:PRINT #6;" T -       - = DAH"
4130 POSITION 1,21:PRINT #6;" U **-"
4150 IF PEEK(53279)<>7 THEN 4200
4160 GOTO 4150
4200 FOR T=0 TO 5
4210 FOR V=200 TO 10 STEP -20
4220 SOUND 0,V,10,5:SOUND 1,V+1,10,5:SOUND 2,V+2,10,5:SOUND 3,V+3,10,5:NEXT V:NE
XT T
4230 SOUND 0,0,0,0:SOUND 1,0,0,0:SOUND 2,0,0,0:SOUND 3,0,0,0:CODE=0:GOTO 3000
5000 LENGTH=0:GRAPHICS 0:SETCOLOR 4,12,7:SETCOLOR 2,0,0:SETCOLOR 1,0,0:POKE 752.
1:RETURN
5010 REM ********** C.S.BARLOW *******
5020 REM ********** C.S.BARLOW *******
5030 REM ********** C.S.BARLOW *******
5040 REM ********** C.S.BARLOW *******
5050 REM ********** C.S.BARLOW *******
5060 REM ********** C.S.BARLOW *******
```

Hi- welcome to the U.K. ATARI COMPUTER OWNERS CLUB software library

This is to be a forum for the exchange of programs, ideas and tips (YOURS as well- NOT just ours !!!)what we propose is thet in each issue we will present a listing of the library's contents - with of course the names of relevant programmers next to them. To obtain a copy of any three of these programs just send us a copy of one of your OWN programs on a cassette or on floppy disc (and include return postage !!) and we will copy this to our library, then I will run off copies of the programs you request (sounds like an offer you cent refuse !)

we are looking for any type of program, be it business, educational or games. Even attractive or intriguing pictures or demo routines will be gladly accepted. What I would particularly like to see is e good flow of machine code or hybrid machine code/BASIC programs with lightning-fast routines for sound and video functions

So, basically, the rules of the game are:-

1) Write yourself a program
2) Chose three programs from our list
3) Send cassette or floppy disk with your program + return postage.
4) Wait patiently behind letter box each morning with your ATARI ready

A few words of warning now.

PLEASE dont send your programs to me on "cheap 'n nasty" cassettes, as I may not have time to verify cassette copies of the programs that you request. Nor do I fancy spending my evenings untangling tape from the inner caverns of my cassette drive !!!!

Programs sent in to me must be original - no copyright materiel (I don't like being woken up in the middle of the night by a knock on the door !!!)

Below is an example of the format that we hope to use:-

Code no:	Title:	Type:	S/D:	Size(K):	Author:

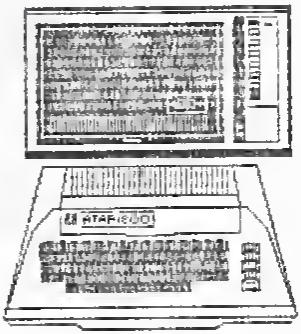
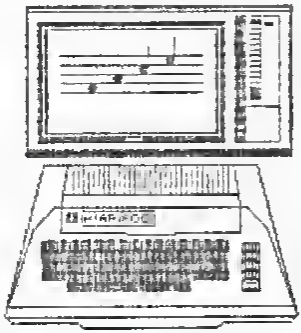
Code No: = A reference number assigned by me

S/D: = Static or dynamic display

Type: = Game, Business, Diepley, Educational

EDITORS NOTE : unfortunately poor old Ron doesnt have any contributions as thie ie hie first issue, so if you send him eomething he will send eomething back, sorry but thie time you will have to take pot luck.Pleaee be generous in your donations as we cant put up with his long face in the office much longer and we figure the only way to make him happy is to give him some programs so that he can make a list.....

Graham

ALL CONTRIBUTIONS,COMMENTS AND SOFTWARE FOR THE PROGRAM EXCHANGE SHOULD BE ADDRESSED TO:-...

UK ATARI COMPUTER OWNERS CLUB,
PO BOX 3,
RAYLEIGH,
ESSEX,
SS6 2BR.